Algoritmi e programmi

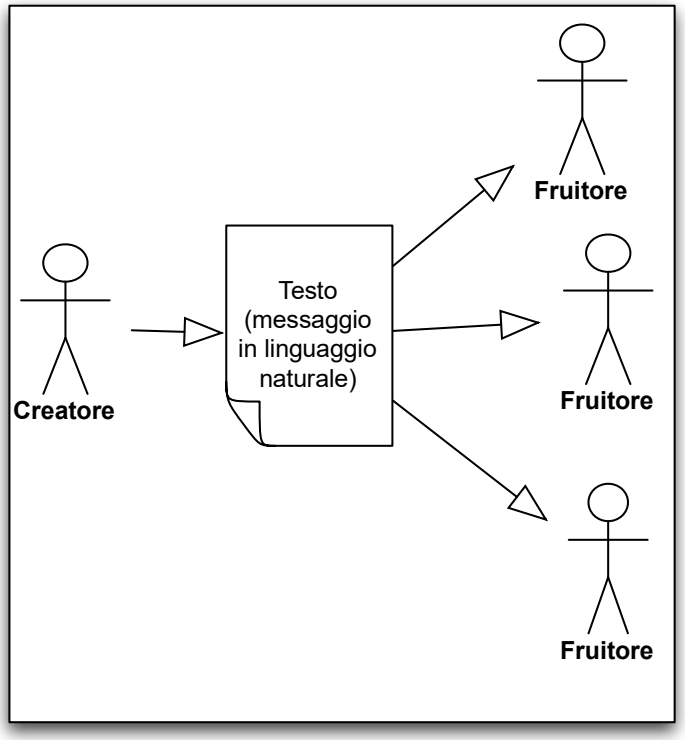
Giovanni Sartor

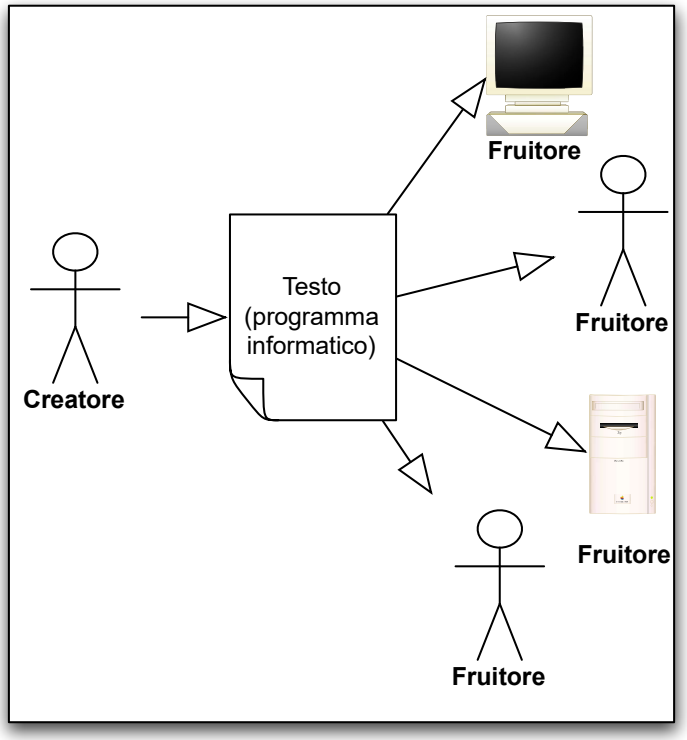
Giuseppe Contissa

E lo spirito divenne carne e prese la sua dimora tra noi

- l'uomo oggettiva se stesso in testi eseguibili automaticamente
- questi danno luogo a realtà virtuali che operano senza l'intervento dell'uomo
- dialettica di necessità e libertà (nuove opportunità all'azione umana, governate da nuovi vincoli)

Testi eseguibili





Ricetta per Mousse di cioccolato

- **Ingredienti:** 250 grammi di cioccolato fondente, 2 cucchiai d’acqua, 60 grammi di zucchero a velo, 6 uova, rum e vaniglia

Ricetta per Mousse di cioccolato

- **Input:** 250 grammi di cioccolato fondente, 2 cucchiai d'acqua, 60 grammi di zucchero a velo, 6 uova, rum e vaniglia
- **Output:** Mousse
- **Procedura:** Fate fondere il cioccolato a bagnomaria con 2 cucchiai di acqua. Quando è completamente sciolto, unite lentamente lo zucchero a velo e poi il burro, un pezzetto alla volta. Mettetelo da parte e sbattete i tuorli per circa 5 minuti, finché non diventano gonfi e spumosi. Uniteli al cioccolato, che avrete leggermente riscaldato se nel frattempo si fosse rappreso troppo. Aggiungete il rum e la vaniglia. Montate a neve ferma gli albumi, aggiungendo poi due cucchiai di acqua. Uniteli con delicatezza al composto di cioccolata e tuorli e versate il tutto in coppette individuali. Fate raffreddare per almeno 4 ore e servite con panna montata a piacere. Le dosi bastano per 6-8 persone.

L’algoritmo, nozione generica

- Che cos’è un algoritmo:
- Un *algoritmo* per un certo esecutore è una combinazione precisa e univoca di azioni, eseguibili autonomamente da parte di quell’esecutore (senza la necessità di un aiuto da parte di altri), che consentono di risolvere un problema.
- Un algoritmo in senso stretto, cioè un algoritmo eseguibile da un esecutore automatico, deve possedere rigorosi requisiti di precisione e univocità che mancano in una ricetta di cucina.
- Un algoritmo presume l’esistenza di un esecutore ma il concetto di algoritmo prescinde dal calcolatore elettronico: una ricetta di cucina è un algoritmo che può essere eseguito da un cuoco, uno spartito musicale viene eseguito da un musicista, ecc.
- Esempio: operazioni necessarie per compiere una telefonata, per prelevare denaro dal bancomat, per iscriversi ad un esame, ecc.

L’algoritmo del bancomat

1. Introdurre la carta magnetica
2. Digitare il codice
3. Selezionare l’opzione Prelievo
4. Indicare l’importo da prelevare
5. Se si desidera lo scontrino premere SI’
6. Altrimenti premere NO
7. Se si era premuto SI’ ritirare denaro, carta scontrino
8. Se si era premuto NO ritirare denaro, carta.

Nell’esempio sono possibile due sequenze di operazioni (FLUSSO CONDIZIONATO):
1, 2, 3, 4, 5, 7
1, 2, 3, 4, 6, 8

Simboli dei diagrammi di flusso

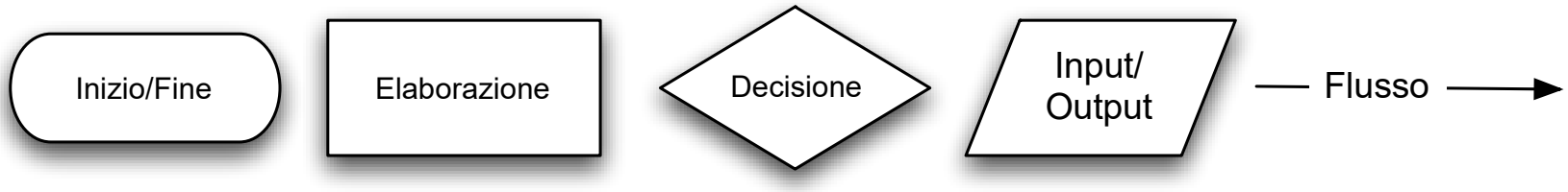
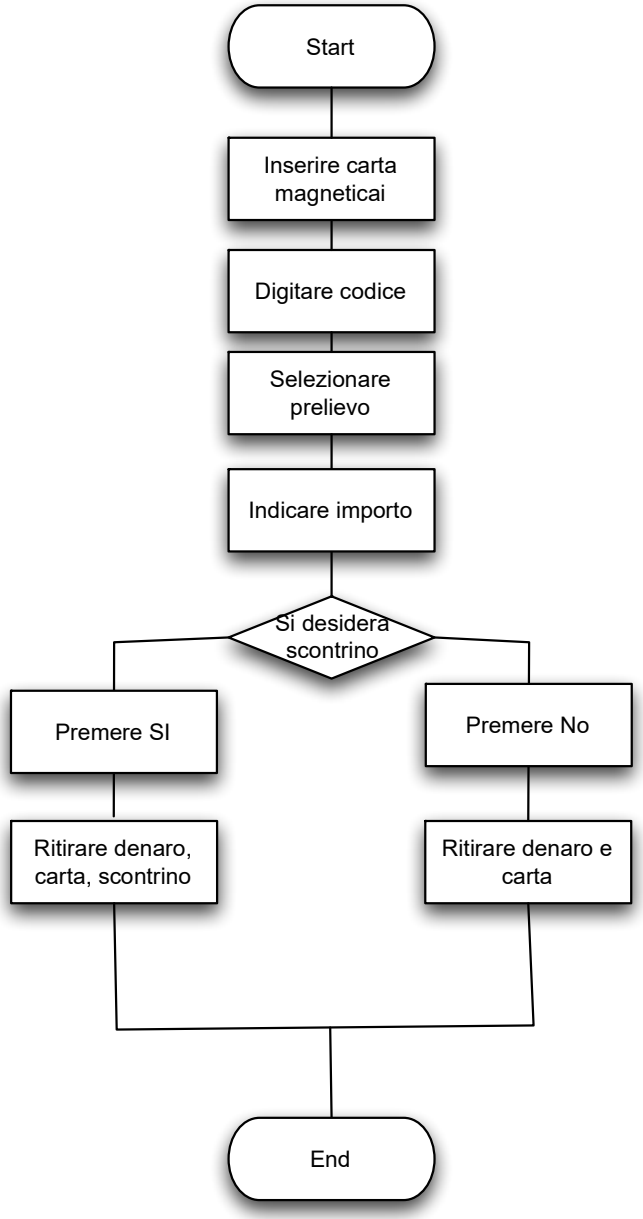


Diagramma del bancomat



Algoritmo per la moltiplicazione in colonna di numeri interi

Input: Due numeri

Output: Il risultato

Procedura:

- si scrivano i due fattori in colonna
- si parta dalla fine del secondo fattore e finché non si siano considerate tutte le cifre di questo si ripeta quanto segue:
  - si moltiplichi il primo fattore per l'ultima cifra non ancora considerata del secondo fattore (partendo dalla fine di questo)
  - si riporti il risultato di questa operazione sotto al risultato precedentemente ottenuto, spostato di una posizione a sinistra (o immediatamente sotto ai fattori, se si tratta del primo risultato)
- infine, si sommino i risultati così ottenuti

		7	8	8	*
			8	9	=
	7	0	9	2	
6	3	0	4	–	
7	0	1	3	2	

Alcune considerazioni sugli algoritmi

- Ogni algoritmo presuppone alcune competenze nell'esecutore (e.g., eseguire le operazioni aritmetiche elementari)
- Mediante algoritmi che usano le azioni elementari si possono svolgere azioni complesse o molecolari (calcolare potenze, radici, funzioni, ecc.); quindi imparando algoritmi si acquista l'abilità di fare nuove cose.
- Gli algoritmi sono generali, riguardando classi di input: l'algoritmo del bancomat è applicabile a qualsiasi cifra da ritirare, l'algoritmo di Euclide o quello della moltiplicazione a qualsiasi coppia di numeri, ecc.
- L'algoritmo è distinto dal processo della sua esecuzione

Non tutti gli algoritmi sono uguali

- Vi sono algoritmi corretti e algoritmi errati.
- Vi sono algoritmi efficienti e algoritmi meno efficienti.

Ricerca sequenziale: prima versione

Input: un elenco ordinato di abbonati al telefono; il nome dell'abbonato da cercare

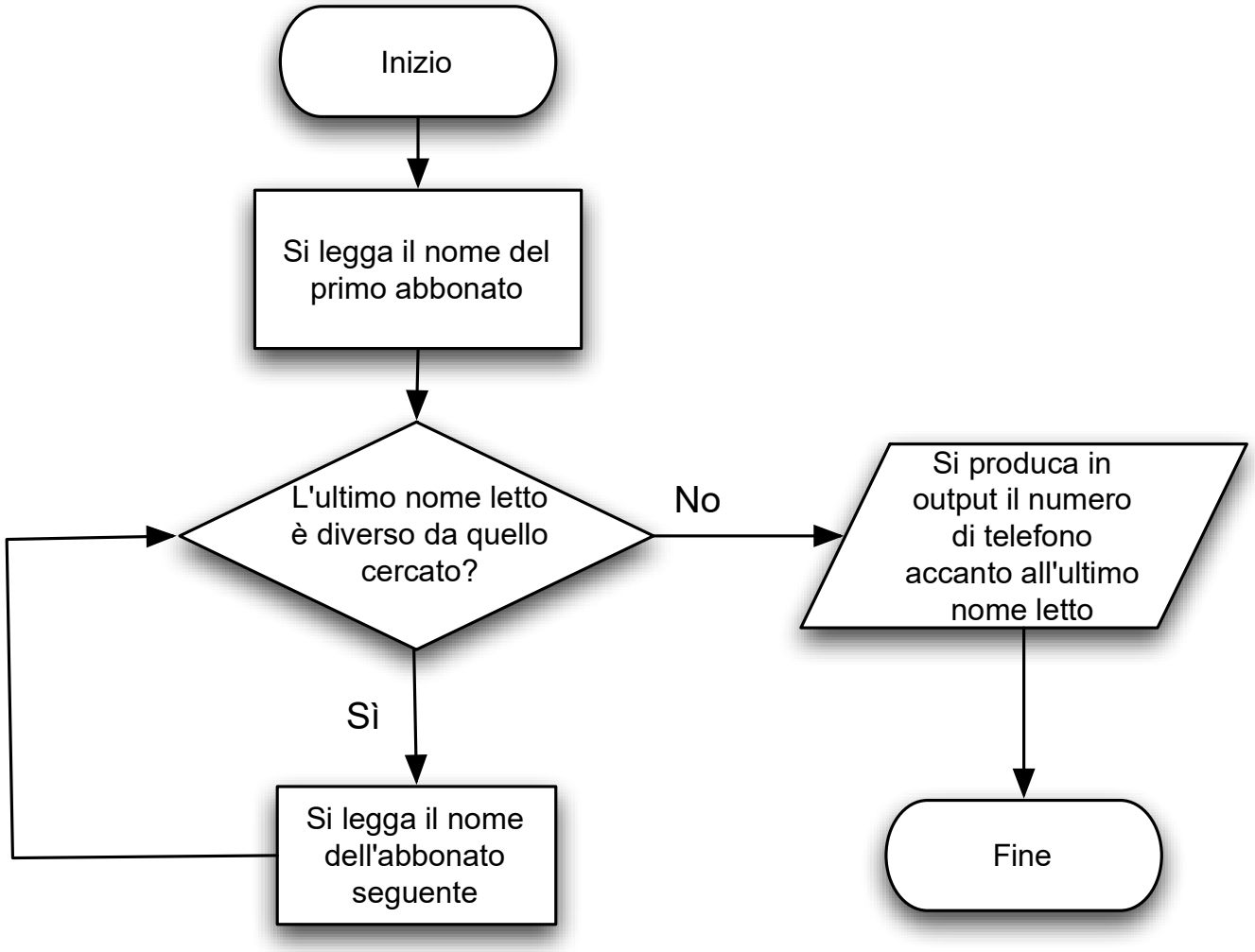
Output: Il numero di telefono dell'abbonato cercato

Algoritmo: Si proceda come segue:

- si legga il nome del primo abbonato dell'elenco,
- finché l'ultimo nome letto è diverso a quello cercato, si ripeta l'operazione seguente:
  - si legga il nome dell'abbonato successivo,
- si annoti come output dell'algoritmo il numero di telefono accanto all'ultimo nome letto.

E' corretto? E' efficiente?

Diagramma di flusso del primo algoritmo per la ricerca



Primo algoritmo per la ricerca

- quanto tempo impiega per cercare un nome in un elenco di 1.000.000 di nomi (1 decimo di secondo per leggere un nome)?
- che accade se un nome non è nell'elenco?

Ricerca sequenziale. Seconda versione

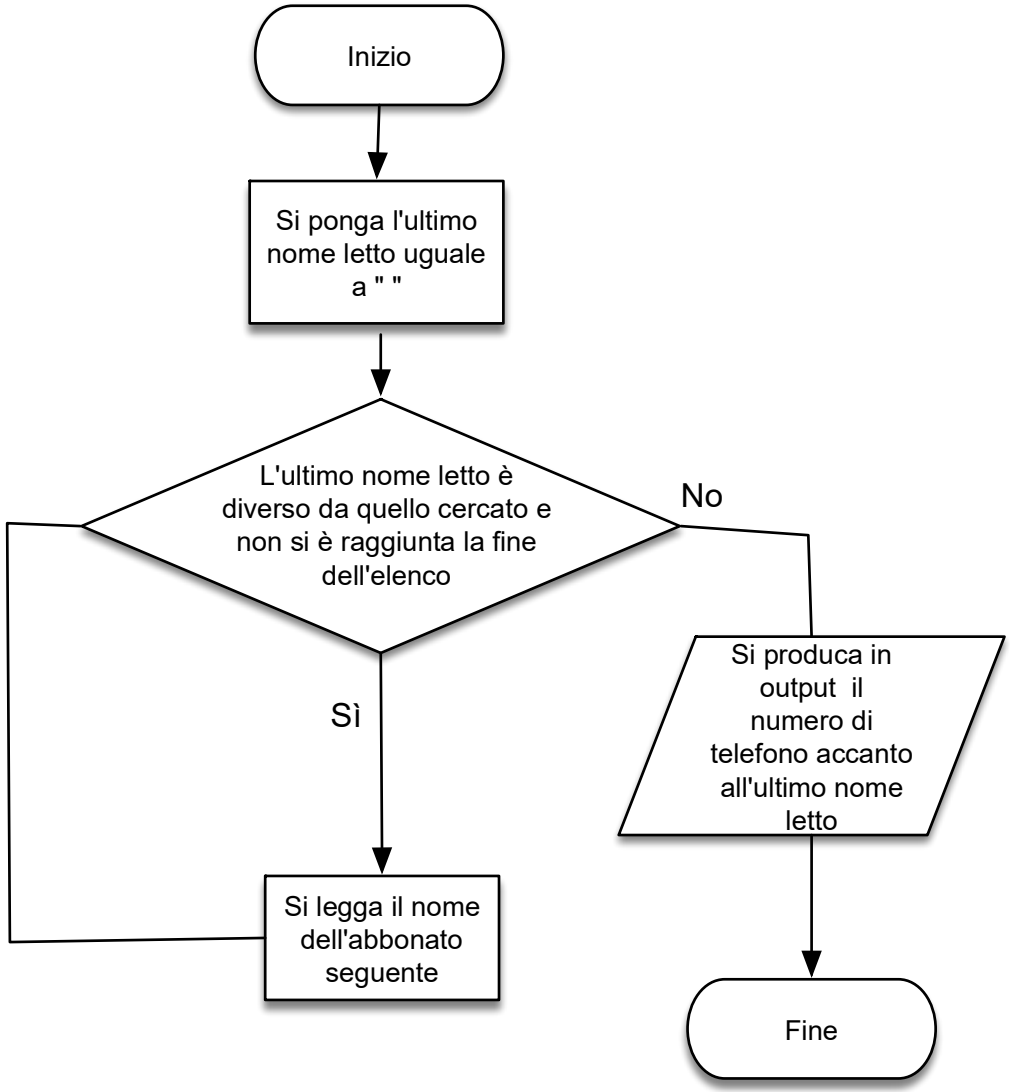
Input: un elenco ordinato di abbonati al telefono; il nome dell'abbonato da cercare
Output: il numero di telefono dell'abbonato cercato

Algoritmo: si proceda come segue:

- si ponga l'ultimo nome letto uguale a " "
- finché l'ultimo nome letto è diverso da quello cercato e non si sia raggiunta la fine dell'elenco, si ripeta l'operazione seguente:
  - si legga il nome dell'abbonato successivo
- si produca in output il numero di telefono accanto all'ultimo nome letto

Che accade se il nome letto non compare nell'elenco?

Diagramma di flusso del secondo algoritmo per la ricerca sequenziale



Ricerca sequenziale. Terza versione

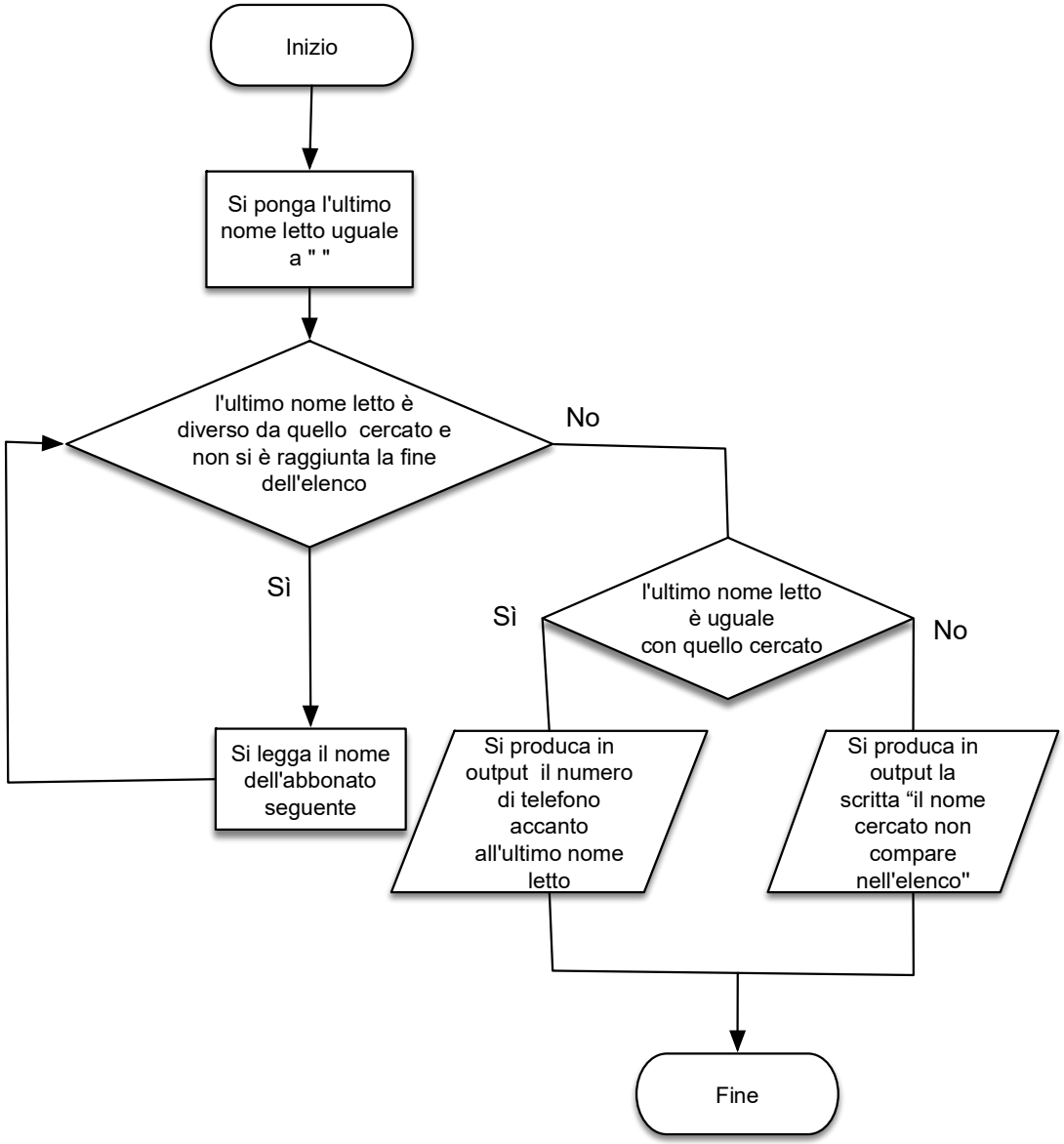
Input: un elenco ordinato di abbonati al telefono; il nome dell'abbonato da cercare

Output: il numero di telefono dell'abbonato cercato

Algoritmo: si proceda come segue:

- si ponga che l'ultimo nome letto sia ""
- finché l'ultimo nome letto è diverso dal nome da cercare e non si sia raggiunta la fine dell'elenco, si ripeta l'operazione seguente:
  - si legga il nome dell'abbonato successivo (che diventa l'ultimo nome letto)
- se l'ultimo nome letto è uguale al nome da cercare, si produca in output il numero di telefono accanto all'ultimo nome letto;
- se l'ultimo nome letto è diverso dal nome da cercare, si produca in output la scritta "il nome cercato non compare nell'elenco"

Diagramma di flusso del terzo algoritmo per la ricerca sequenziale

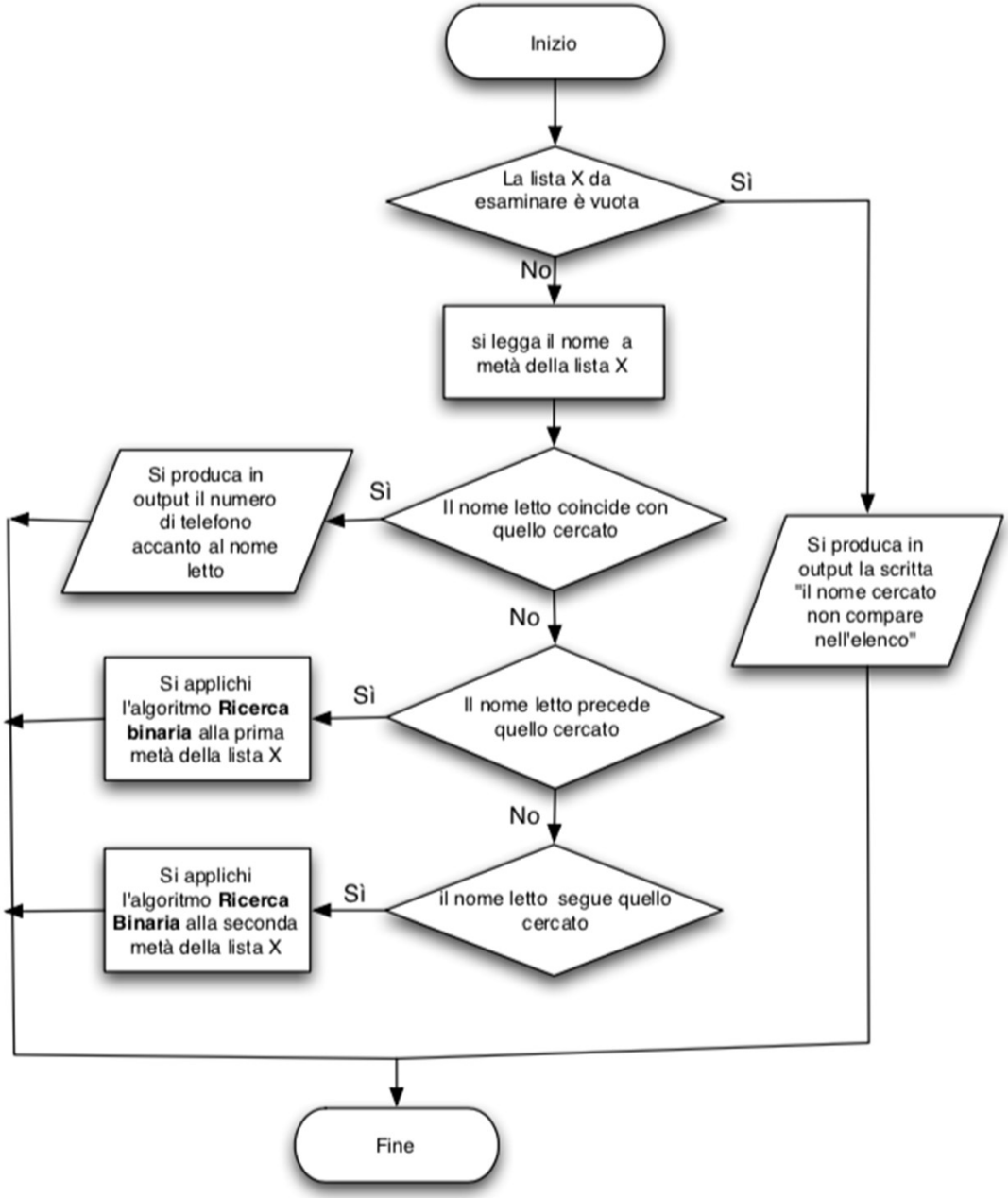


Ricerca binaria (versione ricorsiva)

Input: il nome Y dell'abbonato cercato; una lista ordinata X di nomi di abbonati, ciascuno affiancato dal relativo numero di telefono
Output: il numero di telefono dell'abbonato cercato
Algoritmo: Si proceda come segue:

- se la lista da esaminare non è vuota si proceda nel modo seguente:
  - si legga il nome W dell'abbonato che si trova alla metà della lista X
  - se il nome Y dell'abbonato cercato è eguale a W, allora si produca in output il numero di telefono di W
  - se invece il nome Y dell'abbonato cercato segue W nell'ordine alfabetico, allora si esegua l'algoritmo *Ricerca binaria (versione ricorsiva)* applicandolo all'abbonato da cercare Y e alla seconda metà della lista X
  - se invece il nome Y dell'abbonato cercato precede W nell'ordine alfabetico, allora si esegua l'algoritmo *Ricerca binaria (versione ricorsiva)* applicandolo all'abbonato da cercare Y e alla prima metà della lista X
- se invece la lista da esaminare è vuota, si proceda nel modo seguente:
  - si produca in output la scritta "il nome cercato non compare nell'elenco"

Ricerca binaria ricorsiva: diagramma di flusso



Una definizione precisa di algoritmo

Un algoritmo è una sequenza finita di istruzioni ripetibili e non ambigue. Tale sequenza di istruzioni, se eseguita con determinati dati in ingresso (input), produce in uscita dei risultati (output), risolvendo una classe di problemi in un tempo finito.
Caratteristiche:

- Finitezza: l'algoritmo deve portare alla soluzione in un numero finito di passi.
- Generalità: l'algoritmo non risolve uno solo problema, ma una classe di problemi.
- Non ambiguità: le istruzioni indicate sono specificate univocamente, cosicché la loro esecuzione avviene sempre nello stesso modo, indipendentemente dall'esecutore.
- Ripetibilità (determinismo): dati gli stessi dati in input l'algoritmo deve fornire gli stessi risultati in output.

Non tutti gli studiosi concordano con questa definizione

La correttezza degli algoritmi

Un algoritmo corretto per tutti gli input possibili da sempre la risposta corretta, che risolva il problema sottoposto all'algoritmo. Ma un algoritmo può essere sbagliato:

- Non dare alcuna risposta (pur esistendo una risposta corretta)
- Dare una risposta errata

Esistono tecniche automatiche in grado di individuare alcuni errori algoritmici, molti possono essere evitati da una buona tecnica di programmazione, ma nessuna procedura automatica può individuare tutti gli errori possibili.

Chi è responsabile per l'errore incolpevole?

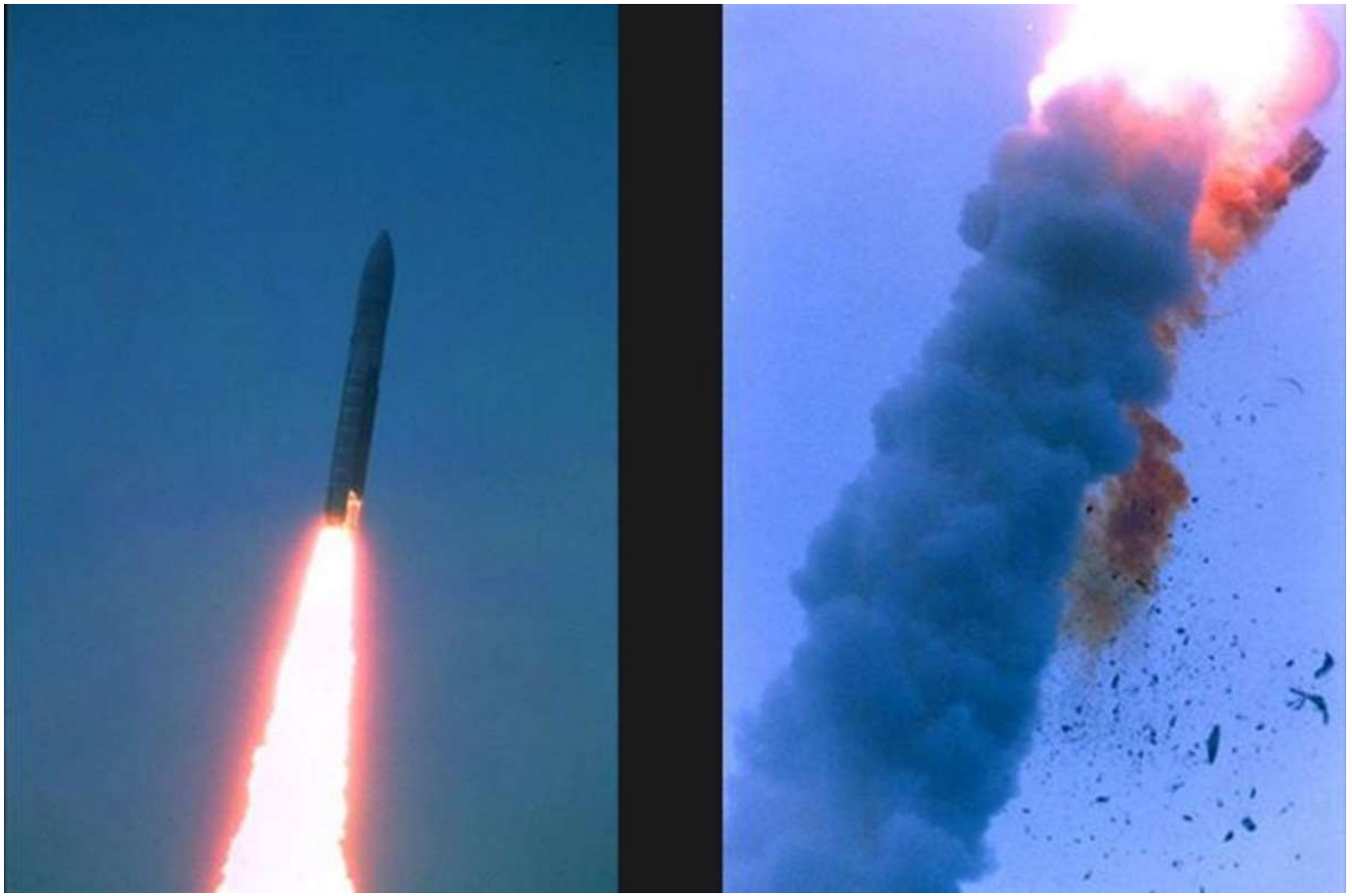
Errore di software: incidente elicottero RAF in Scozia, 1994

- https://en.wikipedia.org/wiki/1994_Scotland_RAF_Chinook_crash#FADEC_problems

Errore di software: incidente aereo Guam, 23 febbraio 2008

- https://en.wikipedia.org/wiki/2008_Guam_B-52_crash

Errore di software: esplosione del razzo Arianne, 1996



- http://en.wikipedia.org/wiki/Ariane_5_Flight_501

Errore di software. L’aereo senza pilota Barracuda (2006)

Efficienza e complessità computazionale

L'efficienza degli algoritmi è misurata dalla loro *complessità computazionale*, cioè dal rapporto tra:

- l'aumento dell'ampiezza dell'input di un algoritmo, e
- l'aumento della lunghezza del processo di esecuzione dell'algoritmo stesso.

Livelli di complessità

- bassa: l'aumento dell'input determina un aumento meno che proporzionale del processo
- media: l'aumento dell'input determina un proporzionale del processo
- alta: l'aumento dell'input determina un aumento più che proporzionale del processo

Algoritmo ad alta complessità computazionale: Individuazione del minimo fattore

Input: un numero da esaminare, prodotto di due numeri primi

Output: l'indicazione che si tratta di un numero primo o altrimenti, l'indicazione del più piccolo divisore intero del numero in esame

Algoritmo: Si proceda come segue:

- il valore iniziale del divisore si ponga eguale a 1
  finché la divisione dà un resto diverso da 0 e il quadrato del divisore è inferiore al numero esaminato
  - si ponga quale divisore il successore del divisore presente (in altri termini, si aggiunga 1 al valore del divisore)
  - si divida il numero in esame per il divisore così calcolato
- se il resto dell'ultima divisione è 0 si annoti che il divisore è un fattore del numero in esame
- se invece il quadrato del divisore è superiore al numero in esame, allora si annoti che il numero in esame è un numero primo

Ogni volta che si aggiungono cifre a un numero da esaminare, aumenta di alcune volte il numero di divisioni da eseguire

La crittografia a chiave doppia: applicazioni basate sui limiti degli algoritmi

La chiave pubblica è il prodotto di due grossi primi, quella privata è data dai fattori.
Dalla chiave pubblica non si può risalire ai fatti perché l'algoritmo della fattorizzazione è ad altissima complessità.

Chi cifra il messaggio usando la chiave pubblica (il prodotto) sa che solo chi possiede la chiave privata (i fattori) sarà in grado di decifrarlo.
Se il messaggio (o una sua parte) è stato invece cifrato usando la chiave privata, chi lo decifra usando la chiave pubblica sa che il messaggio può provenire solo dal possessore della chiave privata corrispondente, e che non è stato alterato (altrimenti la sua decifrazione con la chiave pubblica non sarebbe possibile).

Dagli algoritmi ai linguaggi di programmazione

- Programma: Testo eseguibile da un computer.
- Doppia destinazione: All'uomo (programmatore, analista, studioso), e al computer (esecutore)
- Solo studiando il programma è possibile correggerlo e adattarlo, e comprendere perché il computer si comporta in un certo modo.

Dal linguaggio macchina all'assembler

	codice dell'operazione	indirizzo dell'operando
Linguaggio macchina	010100	001000

Codice dell'operazione 010100, pari al numero decimale 20 , l'indirizzo dell'operando, numero 001000 pari a 8

Assembler	codice dell'operazione	indirizzo dell'operando
linguaggio macchina	010100	001000
Assembler	LOAD	8

Linguaggi ad alto livello rispetto al linguaggio macchina

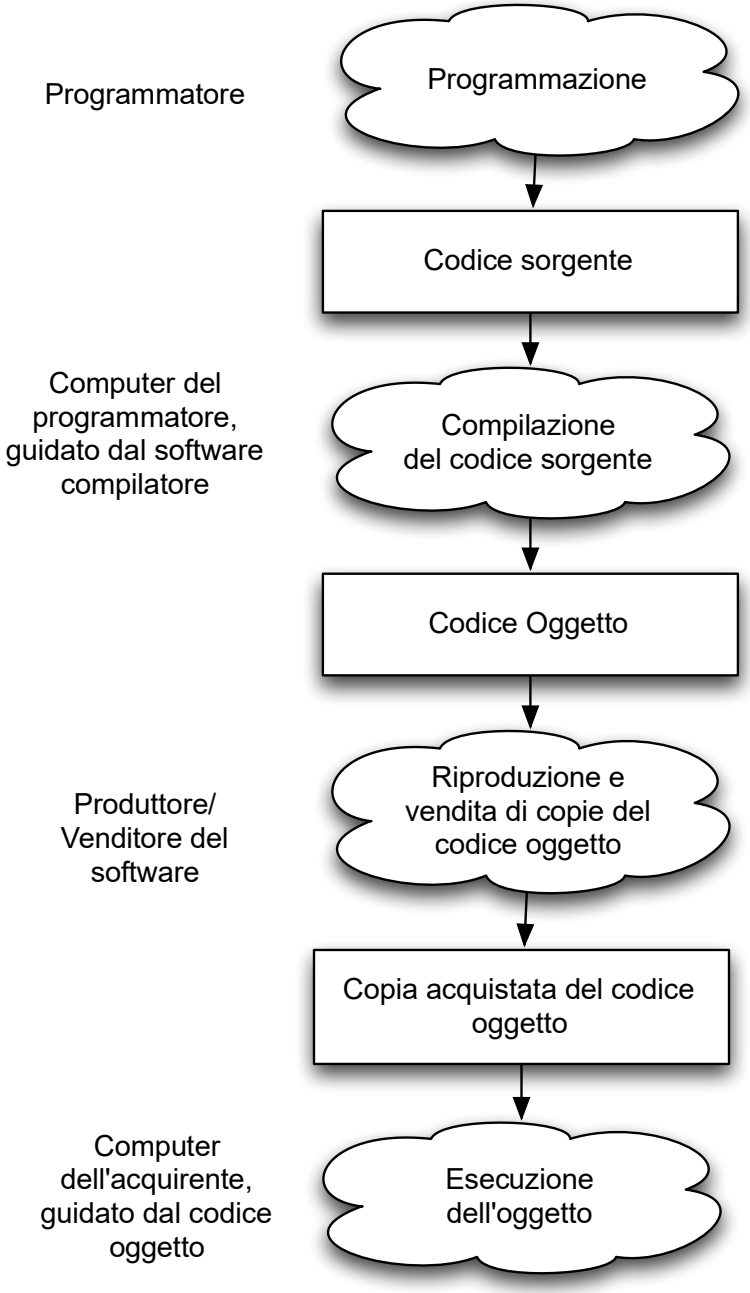
- Più facilmente comprensibili per l'uomo (abbastanza vicini al linguaggio umano);
- Più astratti/compatti;
  Indipendenti da un particolare hardware;
- Debbono essere tradotti nel linguaggio macchina per essere eseguiti dall'hardware.

Esempio di codice in linguaggio di programmazione ad alto livello (BASIC)

```
FUNCTION gcd(a%, b%)
    IF a > b THEN
        factor = a
    ELSE
        factor = b
    END IF
    FOR l = factor TO 1 STEP -1
        IF a MOD l = 0 AND b MOD l
= 0 THEN
            gcd = l
        END IF
    NEXT l
    gcd = 1
END FUNCTION
```

https://rosettacode.org/wiki/Greatest_common_divisor

Compilazione

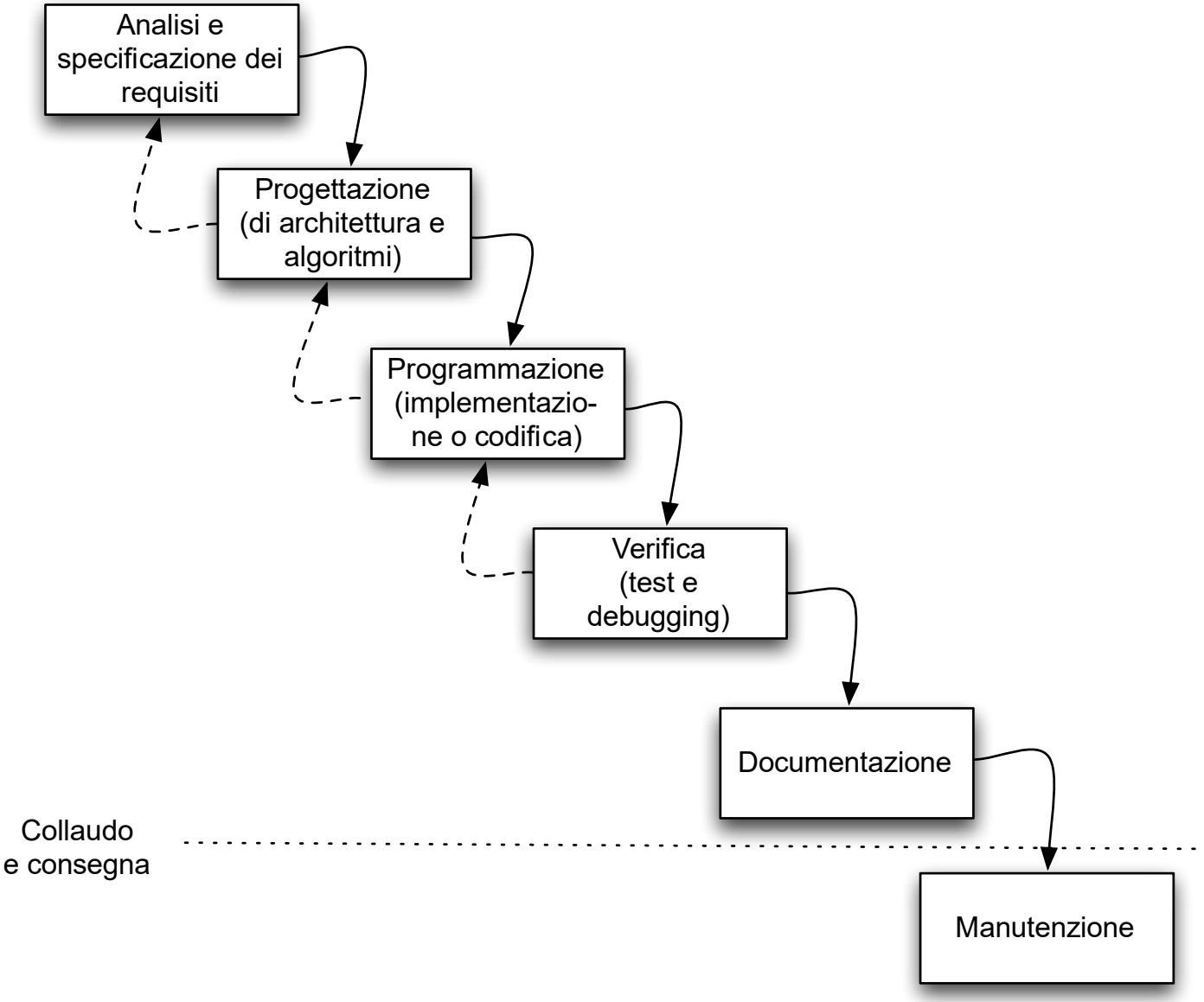


Interpretazione

- si parta dall'inizio del programma sorgente
- finché non si sia giunti alla fine del sorgente, si ripetano le azioni seguenti:
  - si legga l'istruzione successiva del codice sorgente
  - la si traduca in un insieme di istruzioni in linguaggio macchina
  - si eseguano tali istruzioni.

Soluzione intermedia: bytecode (di Java)

Lo sviluppo del software: modello a cascata



Ontologia del software